



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/755,282	01/13/2004	M. George George	200311859-1	1629
22879	7590	07/09/2008		
HEWLETT PACKARD COMPANY P O BOX 272400, 3404 E. HARMONY ROAD INTELLECTUAL PROPERTY ADMINISTRATION FORT COLLINS, CO 80527-2400				
			EXAMINER	ZHE, MENG YAO
			ART UNIT	PAPER NUMBER
			2195	
NOTIFICATION DATE	DELIVERY MODE			
07/09/2008	ELECTRONIC			

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com
ipa.mail@hp.com

Office Action Summary	Application No. 10/755,282	Applicant(s) GEORGE, M. GEORGE
	Examiner MENGYAO ZHE	Art Unit 2195

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
 - If no period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
 - Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 28 March 2008.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-34,37 and 38 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-34, 37-38 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO-166/08)
 Paper No(s)/Mail Date _____
- 4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date _____
- 5) Notice of Informal Patent Application
 6) Other: _____

DETAILED ACTION

1. Claims 1-34, 37-38 are presented for examination.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 15-26 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

A. The following claim languages are unclear and indefinite:

i) Claim 15, line 10, it is uncertain what is meant by "resetting the private variable" <i.e. when was the variable even set? In step 2?

However, step 2 can only happen if the thread enters the code. And what value is the variable set to?>.

Line 11, it is uncertain what is meant by "not set" <i.e. not set to what value? Does it mean that it's in a reset state?>.

Line 14, it is uncertain why step C is needed <i.e. if step A already reset the private variable, why would it be set again?>

Overall, claim 15 is a method claim that recites a sequence of steps, i.e. steps 1-5. However, some steps such as steps 2, 4, 5 are

conditional. Because some steps depend on other conditional steps, it is unclear how those dependent steps would be performed if the conditional steps does not hold true <i.e. step 2 is conditional, but step 4 depends on step 2 in the sense that the private variable has to be reset in step 4.a. after it being set in step 2. However, if step 2 does not hold true, how would step 4 be performed?>. Therefore claim 15 is indefinite due to reasons above. The applicant is suggest to positively recite all the steps in the method claim <i.e. step 2 may read: each thread enters the portion of code and sets the private variable.>.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-34, 37-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Elmendorf et al., Patent No. 7,031,989 (hereafter Elmendorf) in view of Bacot et al., Patent No. 5,235,687 (hereafter Bacot).

5. As per claims 1, 16, 17, 19, 21, 24, 26, 27, 28, 31, 33, 37-38 Elmendorf teaches a method of maintaining data objects which is being accessed by a plurality of threads from an application, including the steps of:

i) creating a plurality of private variables corresponding to the plurality of threads (Column 6, lines 30-32);

ii) setting a delete module variable (Fig 8, unit 830: system freshness indicator corresponds to the variable);

when the private variable is in a reset state, delete the module (Fig 8, unit 855; Column 7, lines 25-31);

wherein the private variable is never in a reset state when the thread is within the module (Column 6, lines 30-35), wherein the use of locks within the performance path of the interface module is not required, and wherein threads are created and destroyed dynamically (Column 1, lines 55-65).

Elmendorf does not specifically teach dynamically creating private variables and thus deregistering the private variables for destroyed threads.

However, because Elmendorf does teach that thread may be created dynamically and a private variable is assigned to each thread (Column 1, lines 55-65; Column 6, lines 30-32), it would have been obvious to one having ordinary skill in the art to see that the private variable is dynamically deregistered once the thread is destroyed since in the art of multithreading, when a thread is destroyed, its associated resources are also destroyed as well, including its private variables.

Elmendorf does not specifically teach a that the module is replaced after being deleted, therefore Elmendorf does not teach a replace module variable, which when set, blocks threads from entering the implementation module.

However, Bacot teaches a replace module variable, which when set, blocks threads from entering the implementation module (Column 11, lines 30-40) for the purpose of preventing incorrect read from the module.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to modify the teachings of Elmendorf with a replace module variable, which when set, blocks threads from entering the implementation module, as taught by Bacot, because it prevents incorrect reads from the module.

6. As per claims 2 and 3, Elmendorf does not specify whether if the implementation module is non-recursive or recursive. However, it would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to have the module be any type of module including non-recursive and recursive, since this does not contribute to the significance or the functioning of the applicant's invention.

7. As per claim 4, Elmendorf teaches wherein each counter is incremented when the corresponding thread enters the implementation module and decremented when the thread exits the implementation module (Column 8, lines 30-37).

8. As per claim 5, Elmendorf does not specifically teach wherein the private variable is in a set state when the value of the counter is above zero and the private variable is in a reset state when the value of the counter is zero or below.

However, it would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to see that the private variable may be set or reset however one chooses, and that setting it when the counter value is above zero is just one obvious way of setting the variable.

9. As per claim 6, Elmendorf teaches wherein each counter is only writable by its corresponding thread (Column 6, lines 29-32).

Elmendorf does not teach that each counter is readable only by its corresponding thread. However, since private registers that only its thread may access or read has been existent at the time of the applicant's invention, it would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to have the counter such that it may only be read by its corresponding thread.

10. As per claim 7, Elmendorf does not specifically teach wherein step (iii) is performed by the interface module. However, it would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to have the interface module to perform step (iii) since the rights of access to the implementation module is controlled by the interface module.

11. As per claim 8, Elmendorf teaches wherein each private variable is modifiable by that variable's corresponding thread (Column 6, lines 29-32).
12. As per claim 9, Elmendorf teaches wherein each private variable is readable by all the threads (Column 6, lines 29-32).
13. As per claim 10, Elmendorf teaches wherein the implementation module is a Library (Column 1, lines 25-32).
14. As per claim 11, 18, Elmendorf does not specifically wherein the private variables and the replace module variable are defined as cache coherent. However, it would have been obvious to one having ordinary skill in the art at the time of the applicant's invention design the variables to be cache coherent so that consistency and validity of all data is maintained.
15. As per claim 12, Elmendorf teaches wherein a thread performs the step (iii) (Column 6, lines 29-32).
16. As per claims 13, 20, Elmendorf does not specifically teach wherein a mutual exclusion primitive is used within step (iii) to ensure that only a single thread performs steps (a) and (b). However, it would have been obvious to one having ordinary skill in

the art at the time of the applicant's invention to have only a single thread performing those steps in order to avoid conflict among multiple threads.

17. As per claim 14, Elmendorf teaches wherein checking of flags within an array is not required in the performance path of the interface module (Column 8, lines 25-57).

18. As per claims 15, 32, 34, Elmendorf teaches a method of synchronizing a plurality of threads for the performance of an action which affects a resource accessed within a portion of code, including the steps of: i) registering for each thread a corresponding private variable (Column 6, lines 30-32); ii) each thread setting the private variable when that thread enters the portion of code (Column 7, lines 3-5); iii) setting a perform action variable when the action is to be performed (Column 6, lines 65-67: the action corresponds to deleting old data); iv) when a thread is within the portion of code and the perform action variable is set, the thread: a. resetting the private variable; b. when the private variables for all threads are not set: i. performing the action (Column 7, lines 34-40: deleting the old data corresponds to the action); c. setting the private variable (Column 7, lines 18-25); and v) when a thread is within the portion of code, the thread: d. using the resource (Column 7, lines 41-45); and e. resetting the private variable (Column 7, lines 25-40); wherein the threads may be dynamically created and destroyed (Column 1, lines 55-65).

Elmendorf does not specifically teach resetting the perform action variable.

However, Bacot teaches resetting the perform action variable (Column 11, lines 50-60; Column 12, lines 55-59) for the purpose of restoring status of modules.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to modify the teachings of Elmendorf with resetting the perform action variable, as taught by Bacot, because it allows for restoration of module status.

19. As per claim 29, Elmendorf teaches wherein the processor is further arranged for resetting the replace module variable when the implementation module has been replaced (Column 8, lines 59-63).

20. As per claim 30, Elmendorf teaches wherein the processor is further arranged for unblocking the threads when the replace module variable has been reset (Column 8, lines 59-63).

21. As per claim 22, Elmendorf does not specifically teach wherein the registration of each private variable occurs when the corresponding thread is created. However, it would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to create all relevant variables related to the thread as soon as the thread is created so that they are available to usage as thread performs its tasks.

22. As per claim 23, Elmendorf does not specifically teach wherein the registration of each private variable occurs when the corresponding thread enters the interface

module for the first time. However, it would have been obvious for one having ordinary skill in the art at the time of the applicant's invention to register the private variable only when it's needed, which happens when its corresponding thread enters the interface module.

23. As per claim 25, Elmendorf does not specifically teach wherein the resource is the kernel or operating system of a machine upon which a process containing the threads is operating and the action is the migration of the process to a new machine. However, Elmendorf does teach that the modules are loaded and unloaded in processing systems dynamically (Column 1, lines 33-40, lines 55-60), therefor it would have been obvious for one having ordinary skill in the art at the time of the applicant's invention to have the modules be loaded onto a new machine, where the machine contain resources such as a kernel since most computer processing systems have kernels and that new programs are always needed to be loaded onto them for different processing purposes.

Response to Arguments

Applicant's arguments filed on 3/28/2008 have been fully considered but are moot in view of new grounds of rejection. However, as for the record, the following is addressed:
In the remark, the applicant argued that:

- i) Pg 15, claims 15, 27 and 28 has additional elements that were not addressed in the previous office action.

The Examiner respectfully disagree with the applicant. As to point:

- ii) Because claim 15 is confusingly stated, the Examiner considers claim 15 to be essentially logically equivalent to claim 1. Claims 27 and 28 are respectfully the interface module and system having components capable of performing the method steps of claim 1. Since claim 1 is rejected, these two claims are rejected based on the same grounds.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MengYao Zhe whose telephone number is 571-272-6946. The examiner can normally be reached on Monday Through Friday, 7:30 - 5:00 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Li B. Zhen/
Primary Examiner, Art Unit 2194